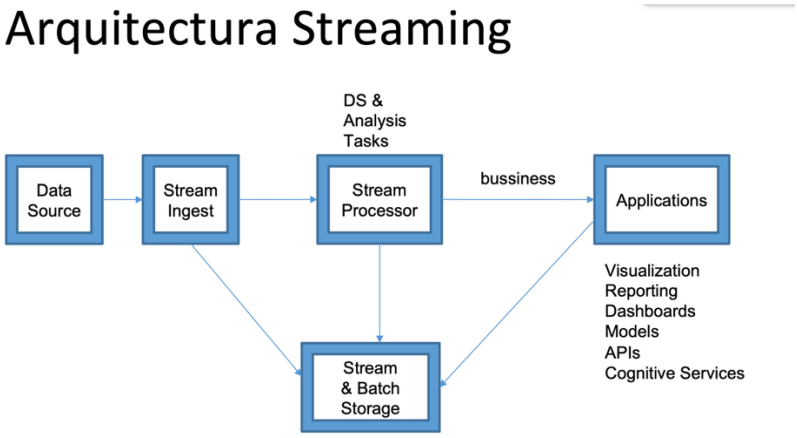
ST1630 Sistemas Intensivos en Datos, 2025-2

Reto2:



FuenteDatos ->
IngestaStreaming->
ProcesadorFlujos->
Almac Datalake / DWH
StreamDB -> App-Visualización
-> Canalizador
-> otra App
DB
Otra

En la sesión 2, se desarrollo la fase conceptual y tecnológica de la arquitectura streaming. Dicha arquitectura contempla fuentes vivas, ingesta en streaming con canalizadores como kafka y en cada nube, almacenamiento en bath y streaming, procesadores de flujo que recibido de los canales de la ingesta puede realizar diferentes aplicaciones sobre los flujos (como aplicaciones conducidas por eventos, ejecutar modelos analiticos, realizar pipelines y etc, entre otros), finalmente, una diversidad de aplicaciones como visualización, notificación, APIs, chatbots, etc.

Durante la sesión 2, se realizó la intrucción a los MOMs, apache kafka, procesadores de flujo como apache flink, introducción al ecosistema ELK, y finalmente la teoria y practica de AWS Kinesis.

Dentro de lo planeado, no se alcanzo a ver la teoria y laboratorio de Spark Streaming, el cual es un procesador de flujos unicamente, y requiere complemento de otras tecnologias como kafka como canalizador, y bases de datos y sistemas de archivos (opción 2). (en los contenidos del curso y en el github hay material de apoyo).

Cada alumno, debe realizar una de las siguientes soluciones end-to-end, es decir, implementando todas las etapas de ciclo de vida de un proyecto de streaming, inspirado en algunos de los ejemplos vistos en clase y material de apoyo.

Cada alumno deberá escoger una opción:

1. Ecosistema completo con Apache Kafka. Utilizar una fuente viva simulada, LogStream es una buen opción, puede inspirarse en el lab de aws kinesis. Bluesky es tambien otra muy buena alternativa para trabajar con redes sociales, es una alternativa a twitter/x que permite tener un API para desarrollor y leer mensajes en vivo. Puede creativamente utilizar otras fuentes vivas como webstream, IoT y simulación de sensores, etc. Utilizar como Canalizador y Procesador de flujos al ecosistema de Kafka. Utilizar bases de datos bath y stream con Apache connect o similar y realizar alguna visualización en tiempo real con Grafana. Tambien podría emplear otras aplicaciones como notificación.

Fuente Viva->
IngestKafka->
KafkaStreams->
Datalake | DB
DBnosql | DBsql | canalizador ->
AppVisualización (ideal con grafana | otras apps

Kafka connect -> salidas a: db, canalizadores, etc.

Ejemplos: https://github.com/st1630eafit/st1630-252/tree/main/sesion2/kafka

2. Ecosistema end-to-end con Apache Spark, usando Apache Streaming.

Es una una variante de la Opción 1, y en vez de utilizar Kafka Stream, se utilizaría Spark Streaming. Explorar posibilidad de introducir Databricks

Ejemplos: https://github.com/st1630eafit/st1630-252/tree/main/sesion2/kafka
https://github.com/st1630eafit/st1630-252/tree/main/sesion2/spark-streaming

3. Ecosistema end-to-end con ELK: Elastic Es un ecosistema muy robusto de implementación de sistemas basados en Streaming. Ha incorporado una nueva ola de aplicaciones modernas basadas en GenIA y LLM. Es una oportunidad de conocer y apropiar este ecosistema, que muchas nubes como AWS están adoptando.

Se recomienda utilizar la versión free-tier en nube que provee elastic.co o versión local n máquinas virtuales nativas o con docker (opción recomendada), y que n puede ser igual a 1, es decir montar todo el reto2 en una sola máquina virtual con docker o nativo.

Se recomienda explorar el ejemplo en el github de la materia, el cual explora alternativas básicas, se recomienda fuentes vivas con Beats con Metrics o File (ver ejemplos del github o el sitio

elastic.co). El sitio web elastic.co cuenta con numerosos ejemplos, tutoriales, seminarios en video, etc para realizar un buen reto2, sea creativo e innovador.

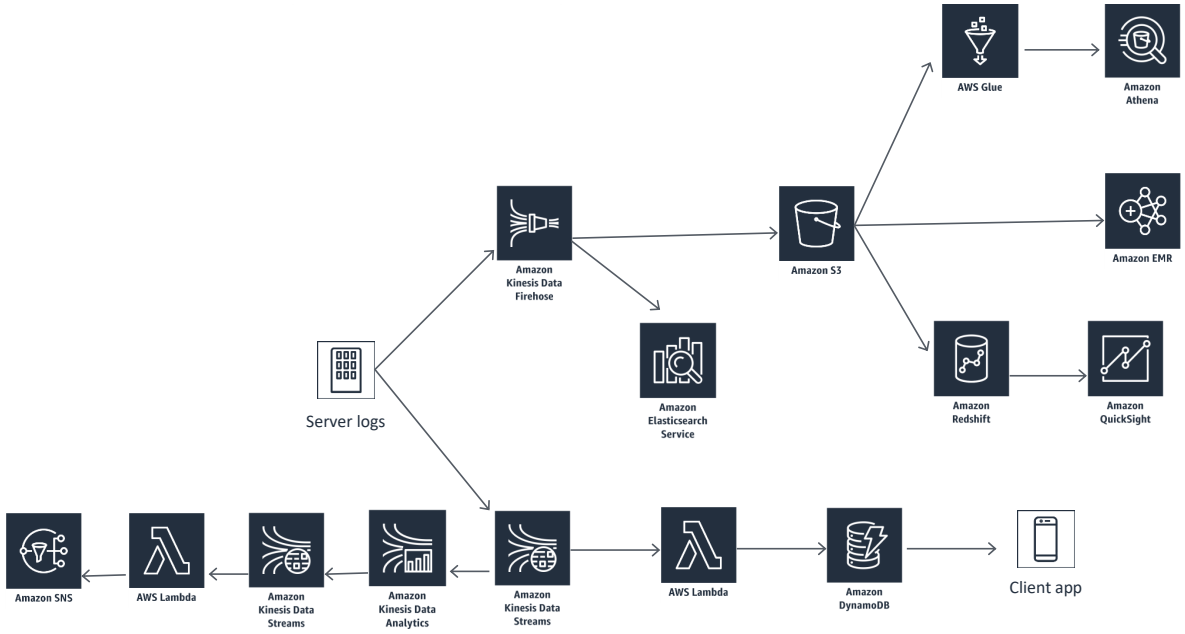
Fuente Viva->
Beats->
->algun-canalizador (ej: kafka)
->logStash
->ElasticSearch
-> procesamiento
->visualización en Kibana

Ejemplos: https://github.com/st1630eafit/st1630-252/tree/main/sesion2/elk

4. Ecosistema end-to-end con AWS Kinesis

Ver demos y material del curso
Utilizar el procesador de flujos AWS Kinesis Flink tanto en SQL como en python.

El demo visto en clase y que el alumno complementará, tendrá que implementar la arquitectura completa de este gráfico:



En la clase de la sesión2, se realizó el demo y grabación de:
Server Logs -> Kinesis Firehose -> S3 -> Glue -> athena

Ud deberá completar los siguientes escenarios:

Server Logs -> Kinesis Data Stream -> AWS Lambda -> DynamoDB

Server Logs -> Kinesis Data Stream -> Kinesis Data analytics (con Kinesis Flink) -> Kinesis Data Stream -> AWS Lambda -> AWS SNS.

Este ultimo caso, implementará un procesador de flujos simulando una aplicación event-driven, en el cual, cuando el inventario de un producto este por debajo de un valor, debe enviar una notificación via SMS (se usa el servicio AWS SNS).

El procesador de flujos, debe ser implementado en SQL y python.

Ver recursos del github: https://github.com/st1630eafit/st1630-252/tree/main/sesion2/kinesis

Ver video de la clase sesion2-kinesis-demo:
Recap: sesion2-labs 31 December | Meeting | Microsoft Teams

5. Ecosistema end-to-end con GCP: inspirado principalmente en el ejemplo de clase de AWS Kinesis, y de la opción 4 de estos retos, realizar la migración a los servicios equivalente en Google GCP, no se limite por las opciones de la opción 4, sea creativo e innovador, si encuentra otros ejemplos o demos en la plataforma GCP, realizarlos, si utiliza otros componentes e integraciones diferentes a las planteadas en la opción 4 realizarlos, por ejemplo: otras fuentes de datos, bases de datos nosql, procesadores de flujo, aplicaciones, etc.

6. Ecosistema end-to-end con Azure: inspirado principalmente en el ejemplo de clase de AWS Kinesis, y de la opción 4 de estos retos, realizar la migración a los servicios equivalente en Microsoft Azure, no se limite por las opciones de la opción 4, sea creativo e innovador, si encuentra otros ejemplos o demos en la plataforma Azure, realizarlos, si utiliza otros componentes e integraciones diferentes a las planteadas en la opción 4 realizarlos, por ejemplo: otras fuentes de datos, bases de datos nosql, procesadores de flujo, aplicaciones, etc. Explore alternativas con Azure Databricks que incorpora todo un ecosistema basado en Apache Spark, podría utilizar el procesador de flujos basado en Spark Streaming (ver opción 2).